Scientometrics (2014) 101:477–504
DOI 10.1007/s11192-014-1375-7

How to kill inventors: testing the Massacrator© algorithm for inventor disambiguation

Michele Pezzoni · Francesco Lissoni · Gianluca Tarasconi

Received: 25 September 2013 / Published online: 29 July 2014

Abstract Inventor disambiguation is an increasingly important issue for users of patent data. We propose and test a number of refinements to the original Massacrator algorithm, originally proposed by Lissoni et al. (The keins database on academic inventors: methodology and contents, 2006) and now applied to APE-INV, a free access database funded by the European Science Foundation. Following Raffo and Lhuillery (Res Policy 38:1617–1627, 2009) we describe disambiguation as a three step process: cleaning&parsing, matching, and filtering. By means of sensitivity analysis, based on Monte-Carlo simulations, we show how various filtering criteria can be manipulated in order to obtain optimal combinations of precision and recall (type I and type II errors). We also show how these different combinations generate different results for applications to studies on inventors' productivity, mobility, and networking; and discuss quality issues related to linguistic issues. The filtering criteria based upon information on inventors' addresses are sensitive to data quality, while those based upon information on co-inventorship networks are always effective. Details on data access and data quality improvement via feedback collection are also discussed.

Keywords Patent data · Inventors · Name disambiguation

M. Pezzoni (✉)
CEMI, École Polytechnique Fédérale De Lausanne, Odyssea, 1015 Lausanne, Switzerland
e-mail: michele.pezzoni@epfl.ch

M. Pezzoni · F. Lissoni · G. Tarasconi
CRIOS, Università Bocconi, Via G. Roentgen 1, 20136 Milan, Italy
e-mail: francesco.lissoni@u-bordeaux.fr

G. Tarasconi
e-mail: gianluca.tarasconi@unibocconi.it

F. Lissoni
GRETHA UMR 5113, Université de Bordeaux, Avenue Léon Duguit, 33608 Pessac cedex, France

JEL Classification C15 · C81 · O34

Introduction

Economic studies of innovation have for long made use of patent data (Griliches 1990; Nagaoka et al. 2010). Assisted by digitalization of records and increasing computational power, economists and other social scientists have extracted increasing quantities of information from patent documents, such as the applicants' identity and location, the technological contents of the invention, or the latter's impact, as measured by citations. More recently, information on inventors has attracted a good deal of attention. Identifying inventors allows studying their mobility patterns, both in space and across companies (Agrawal et al. 2006; Marx et al. 2009) as well as their social capital, as measured by their position in co-inventor networks (Fleming et al. 2007; Breschi and Lissoni 2009; Lissoni et al. 2010). Inventor data can also be matched to additional information at the individual level, ranging from professional identities (does the inventor appear also on a list of R&D employees? or a list of academic scientists?) to other type of archival data on knowledge-related activities (such as scientific publications; see Azoulay et al. 2009; Breschi et al. 2008; Lissoni et al. 2008).

Identifying inventors within any given set of patent data, as well as matching them to any other list of individuals, requires the elaboration of complex "disambiguation" algorithms. They are necessary to analyse in a non-trivial way the text strings containing the inventors' names, surnames, and addresses. Yet, it is only of late that users of inventor data have started discussing openly about the disambiguation techniques they employ, and examine their implications in terms of data quality and reliability of the evidence produced (Raffo and Lhuillery 2009; Li et al. 2014).

This paper describes and comments upon Massacrator© 2.0, the disambiguation algorithm we elaborated to create the APE-INV inventor database, an open-access initiative funded by Research Networking Programme of the European Science Foundation (http://www.esf-ape-inv.eu). The APE-INV inventor database has been conceived as a subset of the PatStat-CRIOS database (http://db.crios.unibocconi.it/), which contains all patent applications filed at EPO, as derived from the October 2013 release of the Worldwide Patent Statistical Information Database (better known as PatStat[1]). As such, it can be more generally described as a PatStat-compatible dataset, which addresses the needs of the increasingly large community of PatStat users.

Massacrator© 2.0 is a revised form of the original Massacrator© algorithm, which was conceived for the *ad hoc* purpose of identifying inventors in selected countries, and with the intent of maximizing precision (that is, minimizing type I errors, or false positives, Lissoni et al. 2006). Our revision has transformed it into a more general tool, one susceptible of further improvements and that users can calibrate also to maximize recall (minimize type II errors, or false negatives) or to obtain any Pareto-optimal combination of recall and precision (that is, to strike a balance between different types of errors).

In what follows, we first review the relevant literature on disambiguation of inventors (section. Background literature). We then describe the general workflow (cleaning & parsing → matching → filtering) of the Massacrator© 2.0 algorithm (section An overview of Massacrator© 2.0). Then, in section Cleaning & Parsing, we present our calibration

[1] Access information for PatStat at: http://forums.epo.org/epo-worldwide-patent-statistical-database/ - last visited: 6/27/2014.

methodology for the filtering step, which crucially affects the algorithm's performance. In section Matching methodology, we perform a validation exercise, based on two "benchmark" datasets. In the same section, we move on to apply the validated algorithm to the entire PatStat data, in order to generate the APE-INV inventor database. Section Filtering concludes.

Background literature

Name disambiguation (also known as name or entity resolution) is an important operation within most text mining processes. It consists in assigning a unique identifier to $n > 1$ records containing information on the same entity (for example, a patent assignee or inventor). Such operation can be performed in many different ways, which Torvik and Smalheiser (2009) classify in four groups, (i) hand-checking of the individual identity; (ii) wiki-type efforts based on the voluntary contribution of a community of data users; and algorithm-based approaches, either (iii) supervised or (iv) unsupervised. We focus here on algorithms.

Following Raffo and Luhillery (2009), a disambiguation algorithms can be described as a three step process:

1. *Cleaning & Parsing*: the relevant text strings (in our case, those containing information on name, surname and address of the inventor) are purged of typographical errors, while all characters are converted to a standard character set. If necessary, any relevant string is parsed into a several substrings, according to various criteria (punctuation, blank spaces, etc.). Typically, the string containing the inventors' complete name (e.g. Duck, Prof. Donald) is parsed into name, surname and title (if any). The address is parsed too.
2. *Matching*: the algorithm selects pairs of entities who are likely candidates to be the same person, due to homonymy or similarity of their names (in our case, the entities are inventors of different patents).
3. *Filtering*: the selected pairs are filtered according to additional information retrieved either from the patent documentation or from external sources. Some typical information from within the patent documentation is the address (e.g. namesakes sharing the same address are believed to be the same person) or some characteristics of the patent. The latter included the patent applicant's name (e.g. homonyms whose patents are owned by the same company may be presumed to be the same person) or its technological contents (as derived from the patent classification system or patent citations).

Supervised and unsupervised algorithms differ in that the former make use of hand-checked subsets of disambiguated observations, which act as "training" sets for the algorithm, while the latter do not. The "training set" must not contain any false positive or negative.

While the cleaning and parsing step are relatively trivial, the matching and filtering operations may be extremely complex, in terms both of the criteria to be followed and of computational c requirements (see Smalheiser and Torvik 2009, for a general review of the literature; and Bilenko et al. 2006, and On et al. 2005, for specific articles on the matching step).

Concerning the matching step, a consensus exists on the need to avoid trivial matching strategies such as the exact match of last and first names, or initials. This would lead to the

exclusion of a non-negligible share of miss-spelled surnames and of the occasional cases of inversion of name and surname. Better use less naive matching strategies, based on various measures of lexical distances between the text strings containing names and surnames, which allow casting a wider net and produce large number of entity pairs, to be passed on to the filtering step. We come back to such strategies in section Matching methodology.

As for the filtering step, the literature focuses on two main issues:

(1) how to relate each disambiguation exercise to a theoretical background, one that takes into account both the sources of errors in the data and how they may interact with the use one wish to make of such data.
(2) how to extract information from metadata in order to find the best criteria for filtering out negative matches or confirming positive ones.

On the theoretical ground, several scholars propose a Bayesian approach, which avoids making arbitrary parametric assumptions on the weights to be assigned to the filtering criteria (Torvik et al. 2005; see Carayol and Cassi 2009, and Li et al. 2014, for applications to inventor data). A related argument is that many parametric approaches treat filtering criteria as independent (and therefore add them one to another when calculating similarity between paired entities), when in fact this is not necessarily the case. For example, proximity in the physical and technological space may be correlated (due to agglomeration effects), but several algorithms add them up when deciding whether two inventors on different patents may be the same person.

An alternative route towards avoiding making such arbitrary assumptions consists in applying appropriate statistical methodologies based on the identification of latent unobserved (uncorrelated) variables, or simply in identifying several groups of criteria that can be more safely assumed to be independent, and then add up filters only when they come from different groups (Smalheiser and Torvik 2009). For example, one can run simulations that randomly assign some weights to the available filtering criteria, each set of weight corresponding to a *de facto* different algorithm (especially when admitting weights equal to zero, which amounts to excluding a filter as irrelevant). By computing the precision and recall[2] results of each simulation against a training set, one can then detect which sets of weights (algorithms) perform better, and retain only a few of them, or just one. Indeed, this is the approach we follow with Massacrator 2.0.

An overview of Massacrator© 2.0

Massacrator 2.0 is a supervised algorithm with peculiar and distinctive features both in the matching and in the filtering step.

Disambiguation of inventors consists in assigning a unique code to several inventors listed on different patents who are homonyms or quasi-homonyms, and share of a set of similar characteristics (e.g. they have the same addresses or patents with the same technological content). Inventors with same code are then treated as one individual.

We apply it to 3,896,945 inventors listed on the EPO patent applications contained in the October 2013 version of PatStat, and implement the three steps as follows:

[2] For the definition of "precision" and "recall", see section Cleaning & Parsing.

Cleaning & Parsing

C&P step 1: characters from an *ad hoc* list are removed, as well as punctuation and double blanks. All remaining characters are converted into plain ASCII.[3] As a result, a new field is created ("Inventor's name"), which contains the inventor's surname (possibly composed of several words, as it happens, for example, with Spanish surnames) and all of his/her names (including second, third or fourth names, and suffixes, such as "junior", "senior", "III" etc.). Similar steps are followed to create the following fields: "Inventor's address" (street's name and the number), "Inventor's city", "Inventor's county", "Inventor's region", and "Inventor's state" (to be intended as sub-national units, as in federal nations such as the US or Germany). "Inventor's country" is derived directly from PatStat (ISO_3166-2 country codes).

C&P step 2: The original "Inventor's name" string from PatStat is parsed in as many substrings as the number of blanks it contains plus one. In the remainder of the paper we will refer to these substrings as "tokens". Due to EPO's conventions in reporting surnames and names, we can safely assume that the first token always contains the inventor's surname (or part of it, in case of double or triple surnames), while the last one always contains the given name (or part of it, in case of multiple names). Most cases are easy to manage since they are written in the form "surname, name" so using comma as separator we can easily parse different components.

Substrings whose contents matches a list of surname prefixes (for example, "De" as found in Dutch, French or Italian surnames) are re-joined to the Surname string. Substrings whose contents matches a list of personal titles (such as "Professor" or "Prof.") are stored in a field different from the name (intitle).

Matching methodology

Massacrator 2.0 matches not only inventors with identical names, but also inventors with similar names, such as those hiding minor misspellings (ex.: "Duck, Donald" and "Duck, Donnald") as well as those resulting from the omission or inversion of words within the name or surname (ex.: "Duck, Donald D." and "Duck, Donald" or "Duck, D. Donald"), for a total of about ten millions matches. In order to do so, it mixes the Token approach just described with an edit distance approach, in particular one based upon the 2-gram (2G) distances.

In detail, the algorithm sorts alphabetically all the tokens extracted from the original PatStat inventor's name text strings, without distinguishing between surnames and names (for a total of almost half a million tokens; tokens of two letters or less are discarded). It then computes the 2G distance between consecutive tokens (e.g. tokens appearing in row n and $n + 1$ in the sorted list). The 2G can be described as the vector distance between two strings of different lengths, normalized by the total length of the strings. In our case it will be:

$$2G\,(t1,\,t2) \;=\; \frac{\sqrt{\sum_{i=aa}^{zz(N)} (G1_i \;-\; G2_i)^2}}{num(t1) \;+\; num(t2)} \qquad (1)$$

where:

[3] See the post "Converting patstat text fields into plain ascii" on the RawPatentData blog (http://rawpatentdata.blogspot.com/2010/05/converting-patstat-text-fields-into.html; last access: March, 2014).

- $G1_{\psi\psi}$ and $G2_{\psi}$ are the number of occurrences of the *i-th* bigram appears in tokens *t1* and *t2*, respectively.
- *num(t1)* and *num(t2)* are the number of characters in tokens *t1* and *t2*, respectively.
- *N* is the number of possible combinations of two consecutive letters (bigrams) in the alphabet of choice (in our case, plain ASCII, from which $N\psi = 650$)[4]

Once all $2G(t_1,t_2)$ distances are computed, consecutive tokens can be assigned to "groups", on the basis of their reciprocal distance, as follows:

- Starting from the top of the token list, token in row 1 is assigned to group 1.
- Then token in row 2 is also assigned to group 1 if its 2G distance from token in row 1 is less than or equal to an arbitrary threshold value $\delta\psi$ (in the case of Massacrator 2.0: $\delta = 0.1$); otherwise the algorithm creates a new group (group 2).
- The algorithm then proceeds in a similar fashion for all rows n and $n + 1$.

Once all groups are defined, the algorithm substitutes to each token the number of its corresponding group. As a result, each "Inventor's name" string is now replaced by a vector of numbers, each of which corresponds to a group of tokens. Any pair of inventors whose "Inventor's name" string contains identical group numbers (no matter in which ordered) are then treated as a match. In case the "Inventor's name" string are composed of a different number of tokens, the minimum common number of tokens (groups) is considered (see Fig. 1 for a practical example). All matches obtained in this way are then passed on to the filtering step.

Filtering

For each pair of inventors in a match, Massacrator calculates a "similarity score", based upon a large set of weighted criteria. By comparing this score to a threshold value (*Threshold*), Massacrator then decides which matches to retain as valid (positive matches), and which to discard (negative matches). The criteria considered are 17, grouped in six families: *network, geographical, applicant, technology, citations,* and *others*. Several criteria are derived from the original Massacrator© and they are quite intuitive, so we do not discuss them (see Table 1[5] for a short description). We discuss instead the approximated structural equivalence (ASE) criterion, which is not present in the original Massacrator© and is rather complex.

The concept of structural equivalence was first introduced to social network analysis by Burt (1987). ASE adapts it to networks of patent citation and it was first proposed by Huang and Walsh (2011) as a method for inventor disambiguation.[6] The basic intuition is that the higher the number of citations two patents have in common, the higher the probability that any two inventors of such patents are the same person. Consider inventors *I*'s and *J*'s sets of patents:

[4] As an example, consider token "ABCABC" as *t1* and token "ABCD" as *t2*. The bigram sets for *t1* and *t2* will be respectively: (AB,BC,CA,AB,BC) and (AB,BC,CD). Applying Equation 1 returns:

$$2G(t1,\, t2) = \frac{\sqrt{(2-1)^2_{AB} + (2-1)^2_{BC} + (1-0)^2_{CA} + (1-0)^2_{CD}}}{5+3}$$

[5] For a definition of patent family, see Martinez (2011).

[6] Huang et al.'s original formula was proposed to compare inventors with no more than one patent each. We have adapted it to the case of inventors with multiple patents.

Inventor's Name
Donald Duck
Donnald D.
Duck Donald Rockerduck
M. Mouse

Row	Token	2-G	Group
1	Donald	-	1
2	Donnald	0.09	1
4	Duck	0.33	2
5	Mouse	0.37	3
6	Rockerduck	0.30	4

Matched inventor's names		Shared groups
Donald Duck	Donnald D.	1
Donald Duck	Duck Donald Rockerduck	1 and 2
Donal D.	Duck Donald Rockerduck	1

Fig. 1 Example of Massacrator "mixed" matching rule

$$P_i = \{p_{i,1}, p_{i,2}, .., p_{i,N_i}\} \quad P_j = \{p_{j,1}, p_{j,2}, .., p_{j,N_j}\}$$

Consider also P_{cit} as the set of all patents in our dataset receiving at least one citation:

$$P_{cit} = \{p_1, p_2, .., p_N\}$$

P_i (P_j) and P_{cit} are then used to compute matrix D_i (D_j), which has as lines patents in P_i (P_j) and as columns the cited patent in P_{cit}. If a patent in P_i (P_j) cites a patent in P_{cit}, the corresponding element in matrix D_i (D_j) takes value 1 (D_z [p_z, p_{cit}] = 1; $z = i,j$); if no citation occurs, it takes value zero (D_z [p_z, p_{cit}] = 0; $z = i, j$). Intuitively, inventors with similar matrixes are more likely to be the same person (their patents cite the same patents).

Massacrator then calculates weights W_{Citing} and W_{Cited}. The former is the inverted number of citations p_i (p_j), that is the inverse of the number of citations received by patent p_{cit} element. These weights allow to give less importance to matrix elements D_z [p_z, p_{cit}] = 1 ($z = i,j$) corresponding to "popular" patents (that is, patents sending out and/or receiving many citations).

Finally the algorithm divides the resulting index by the sum of $num(P_i)$ and $num(P_j)$ in order to normalize for the total number of patents of each inventor in the [I,J] pair (Eq. 2)

Table 1 Description of the criteria and classification in five families of filtering criterion

No.	Name of the criterion [*names of variables in squared brackets*]	Description
	Network	This family of criteria bases on the intuition that two matched inventors who turn out to be socially close are more likely to be the same person. As most patents are invented by two or more inventors, we consider each patent as a social tie between the listed co-inventors (Breschi and Lissoni 2005)
1	Common coinventor [*Coinventor*]	Any two inventors *I* and *J* who have both signed patents with inventor are defined as having a common coinventor
2	3 degrees of separation [*Three degrees*]	Any pair of inventors *I* and *J*, are said to stand at three degrees of separation when at least one of *I*'s coinventor and one of *J*'s coinventor have collaborated on the same patent
	Geographical	This family exploits the inventor's address information
3	City [*City*]	Two inventors share the same city within the address field (e.g. Paris, Rome, Dijon)
4	Province [*Province*]	Two inventors share the same province within the address field (e.g. Cote-d'Or)
5	Region [*Region*]	Two inventors share the same region within the address field (e.g. Bourgogne)
6	State [*State*]	Two inventors have in common the same state within the address field (e.g. Texas)
7	Street [*Street*]	Two inventors share the same street and number within the address field. (e.g. Boulevard Pasteur 32)
	Applicant related variables	This family exploits the characteristics of the patent applicant
8	Applicant [*Applicant*]	Two inventors have signed at least one patent each for the same applicant
9	Small Applicant [*Small applicant*]	As with Applicant, when the applicant has less than 50 inventors affiliated. If this criterion is satisfied also Applicant is satisfied
10	Group [*Group*]	two inventors have signed at least one patent each for two distinct applicants belonging to the same group
	Technology classes	This family of citeria bases on the IPC code that identifies the technology class of a patent. The more digits two codes defining the IPC class have in common, the less the technological distance between the patents. The three criteria in this family are strictly related. In the case inventors *I* and *J* share at least one patent each with 12 digits in common, the other two criteria will be satisfied by definition (they have also 6 and 4 digits in common)
11	IPC 12 [*IPC 12*]	Within the stock of patents attributed to inventor *I* there is at least one patent with 12 digits of IPC code in common with another patent belonging to the inventor's *J* stock
12	IPC 6 [*IPC 6*]	Within the stock of patents attributed to inventor *I* there is at least one patent with 6 digits of IPC code in common with another patent belonging to the inventor's *J* stock
13	IPC 4 [*IPC 4*]	Within the stock of patents attributed to inventor *I* there is at least one patent with 4 digits of IPC code in common with another patent belonging to the inventor's *J* stock

Table 1 continued

No.	Name of the criterion [*names of variables in squared brackets*]	Description
	Citation	This family exploits citation links between patents
14	Citations [*Citation*]	When a patent belonging to the stock of patents of inventor *I* is cited by a patent belonging to the stock of patents of inventor *J*, or vice versa, the pair of inventors has in common one citation
15	Approximated structural equivalence [*ASE*]	Discussed in detail by the end of Sect. Matching methodology
	Others	This family includes two criteria that cannot be classified in all the other four families
16	Rare surname [*Rare surname*]	At least one among the matched inventors' surnames is uncommon within the inventor's country. We identify rare surnames according to the frequency (by country) of first token (which we know to contain surnames) from the "inventor's name" PatStat field
17	Priority date differs for less than 3 years [*Three years*]	A patent's priority dates is the earliest date of application in the patent's family. For each pair of inventors we first calculate the minimum temporal distance (that is, the distance in time between the most recent among inventor *I's* patents and the least recent among *J*'s, or vice versa). The distribution of minimum distances is very skewed, we set a threshold value of 3 years as a filtering criterion (*I* and *J* are more likely to be the same person if temporal distance is less than 3 years)

$$ASE[I,J] = \frac{\sum_{p_{i,1}=p_i}^{p_i,N_i} \sum_{p_{j,1}=p_j}^{p_jN_j} \sum_{p_{cit}=p_1}^{pN} D[p_i,p_{cit}] * W_{citing_{p_i}} * D[p_i,p_{cit}] * W_{citing_p} * W_{cited_{p_{cit}}}}{num(P_i) + num(P_j)} \tag{2}$$

The higher the index, the closer inventors I and J are to the "perfect" structural equivalence (same position in the network of citations).

Massacrator 2.0 find only 291469 non-null $ASE[I, J]$ scores, out of the >10 million matches analysed. The ASE filtering criterion is then considered satisfied by all these matches, no matter the score's exact value.

All the filtering criteria reported in Table 1 are used to compute a similarity score of the matched inventors as follows:

$$\alpha_m = \sum_{i=1}^{17} x_{i,m}$$

where $x_{i,m}$ is a dummy variable that equals 1 if match m meets criterion i, 0 otherwise. The number of retained (positive) matches depends upon the value assigned to the threshold variable (*Threshold*); when the similarity score α_m is larger than *Threshold* inventors in match m are considered to be the same person. This is the most delicate aspect of the algorithm implementation because values assigned arbitrarily can affect strongly the algorithm's performance. For this reason, Massacrator 2.0 relies on a calibration methodology, based upon a MonteCarlo simulation exercise, to which we now move on.

Filtering calibration

The final output of the filtering phase has to consist in a list of inventor pairs:

$$[m, I, J, D_{\alpha_m}], \quad I \neq J.$$

where I and J are the two inventors forming pair m. D_{α_m} is a binary variable that takes value 1 if the two inventors in pair m are believed to be the same person (positive match) and 0 otherwise (negative match), based on their similarity score α_m. and the chosen *Threshold* value. Notice that the output varies according to the number of filtering criteria we decide to use, and the *Threshold* value we set. Calibration serves the purpose of guiding our selection of filtering criteria and *Threshold* value, on the basis of the efficiency of the resulting output.

We measure efficiency in terms of precision and recall:

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

where

$$tp = number\ of\ true\ positives$$
$$tn = number\ of\ true\ negatives$$
$$fp = number\ of\ false\ positives$$
$$fn = number\ of\ false\ negatives$$

We establish whether a positive (negative) match is true (false) by comparing the algorithm's results to information contained in two benchmark databases, namely the "Noise

Added French Academic" (NAFA) and the "Noise Added EPFL" (NAE). Each benchmark database consists of a certain number of inventors, all matched one with the other, plus hand-checked information on whether the match is negative or positive. NAFA contains information on 530 inventors from France, 424 of which result affiliated to a university, the others being homonyms added *ad hoc* for testing purposes (that is, they represent added false positives or "noise"). NAE contains information on 342 inventors, 312 of which are faculty members at EPFL (the Federal Polytechnic of Lausanne, Switzerland), the others being added noise.[7]

For any match in the benchmark datasets we define D_{γ_m} analogously to D_{α_m}. It follows:

$$\begin{aligned} D_{\alpha_m} = 1 \mathrm{U} D_{\gamma_m} = 1 &\Rightarrow \textit{true positives} \\ D_{\alpha_m} = 0 \mathrm{U} D_{\gamma_m} = 0 &\Rightarrow \textit{true negatives} \\ D_{\alpha_m} = 1 \mathrm{U} D_{\gamma_m} = 0 &\Rightarrow \textit{false positives} \\ D_{\alpha_m} = 0 \mathrm{U} D_{\gamma_m} = 1 &\Rightarrow \textit{false negatives} \end{aligned} \tag{3}$$

We expect to observe a trade-off between precision and recall; any identification algorithm can decrease the number of false positives only by increasing the number of false negatives and vice versa. The smaller the trad-off, the better the algorithm. However, to the extent that a trade-off exists, we want to calibrate the algorithm in order to:

- Discard suboptimal sets of filtering criteria, namely those sets which increase recall by decreasing too much precision (and vice versa).
- Choose among optimal sets, according to the research objectives (some of which may require precision to be sacrificed to recall, or vice versa).

We proceed in three steps. First, by means of a MonteCarlo simulation exercise, the algorithm generates a large number of observations, each of which consists of a random set of weights assigned to the filtering criteria, a *Threshold* value, and the corresponding results in terms of precision and recall (*Data generation* step).

Second, the simulation results are split into two sets (*dominant* vs *dominated*), with the dominant results further split into three regions of interest, each of which is characterized by a different mix of precision and recall (*Mapping* step).

Finally, weights are assigned to the filtering criteria, according to the desired results in terms of precision and recall (*Weight calibration*). Notice that weights are binary values (0, 1), which amounts to say that our weight calibration consists in including some filtering criteria (1) and excluding others (0). However, further extensions of Massacrator 2,0 may be conceived, one based on continuous weights (comprised between 0 and 1) or on discrete weights, with top values greater than 1.

Sections Cleaning & Parsing, Matching methodology and Filtering describe in details the three steps.

[7] More precisely, NAFA and NAE contain matches between an inventor and one of his/her patents, and another inventor and one of his/her patents, plus information on whether the two inventors are the same person, according to information collected manually. Having been hand-checked, the matches in the benchmark databases are expected to contain neither false positives nor false negatives. Notice that both NAFA and NAE are based upon the PatStat October 2009 release. A detailed description is available online (Lissoni et al. 2010).

Table 2 Satisfied criteria in benchmark datasets, x^k

	NAE (EPFL)	NAFA (French academics)
City	0.15	0.24
Province	0.01	0.3
Region	0.02	0.42
State	0.02	0
Street	0.04	0.02
IPC 4	0.32	0.31
IPC 6	0.2	0.19
IPC 12	0.1	0.07
Three Years	0.49	0.44
Applicant	0.22	0.25
Small Applicant	0.06	0.03
Group	0.01	0.02
Coinventor	0.09	0.1
Three Degrees	0.13	0.12
Citations	0.08	0.08
Rare Surname	0.07	0.05
ASE	0.07	0.06

Data generation

We generate data for calibration as follows:

1. *Vectors of criteria*: for each pair of inventors m, a set of k dummy variables $x^k_{m\psi}$ ($k = 1\ldots17$) is generated, each of them corresponding to one of the 17 filtering criteria described in Section Background literature.3. x^k_m takes value 1 if the filtering criterion is satisfied at least once by the inventors' pair, zero otherwise. Tables 2 and 3 report the percentage of pairs satisfying each criterion and the resulting correlation matrix.
2. *Vectors of weights and computation of similarity scores*: We draw randomly $W\psi$ vectors of weights from a uniform Bernoulli multivariate distribution, where $W\psi\psi$ is set to 2000. The dimensions of the multivariate distribution are as many as the number of variables in vector x (i.e. $K = 17$). Each draw generates a different vector of weights ω_w, where each *k-th* weight (ω^k_w) can take value one or zero (i.e. binomial weight). Each pair of matched inventors from NAFA and NAE benchmark databases is then weighted as follows:

$$\alpha_{m,w} = x_m \times \omega_w$$

where: w=1 … 2000; m=$\{m_{NAFA}, m_{NAE}\}$; m_{NAFA}=1 …2817; m_{NAE}=1…1011;
and sizes of the matrixes are: $x_m[1 \times 17]; \omega_w[17 \times 1]; \alpha_{m,w}[1 \times 1]$.

Binomial weights can be interpreted as a way to exclude/include randomly the k-th filtering criterion in the x_m set. The product of two vectors $x_{m\psi}$ and $\omega_{w\psi}$ returns in the $\alpha_{m,w}\psi\psi$ similarity score of match m, for a specific set w of weights.

3. *Threshold value* : In order to determine whether a match is positive or negative the algorithm compares each similarity score $\alpha_{m_{NAFA},w}\psi$and $\alpha_{m_{NAE},w}\psi$ to a *Threshold* value.

Table 3 Correlation between criteria k_1 and k_2 corr (x^{k1}, x^{k2},)

	City	Province	Region	State	Street	IPC 4	IPC 6	IPC 12	Three Years	Applicant	Small Applicant	Group	Coinventor	Three Degrees	Citations	Rare Surname	ASE
City	1	0.45	0.3	−0.02	0.18	0.21	0.17	0.12	0	0.28	0.05	0.11	0.15	0.17	0.12	0.05	0.09
Province	0.45	1	0.79	0	0.02	0.21	0.17	0.2	−0.01	0.26	0.02	0.11	0.28	0.28	0.16	0.09	0.12
Region	0.3	0.79	1	0	−0.01	0.15	0.13	0.17	−0.07	0.24	−0.01	0.09	0.25	0.23	0.15	0.09	0.1
State	−0.02	0	0	1	0.02	0.02	0.01	0.03	0.02	0.06	−0.02	0.01	0.02	0.03	0.01	0.06	0
Street	0.18	0.02	−0.01	0.02	1	0.08	0.11	0.05	0.03	0.11	0.08	−0.01	0.03	0.06	0.07	0	0.07
IPC 4	0.21	0.21	0.15	0.02	0.08	1	0.71	0.43	−0.01	0.35	0.16	0.06	0.27	0.27	0.29	0.07	0.26
IPC 6	0.17	0.17	0.13	0.01	0.11	0.71	1	0.6	0.08	0.37	0.19	0.03	0.29	0.29	0.33	0.07	0.31
IPC 12	0.12	0.2	0.17	0.03	0.05	0.43	0.6	1	0.15	0.3	0.16	0.02	0.32	0.3	0.33	0.05	0.3
Three Years	0	−0.01	−0.07	0.02	0.03	−0.01	0.08	0.15	1	0.13	0.08	0.03	0.16	0.16	0.08	−0.06	0.13
Applicant	0.28	0.26	0.24	0.06	0.11	0.35	0.37	0.3	0.13	1	0.34	0.1	0.34	0.37	0.31	0.07	0.3
Small Applicant	0.05	0.02	−0.01	−0.02	0.08	0.16	0.19	0.16	0.08	0.34	1	−0.01	0.16	0.16	0.19	0.02	0.19
Group	0.11	0.11	0.09	0.01	−0.01	0.06	0.03	0.02	0.03	0.1	−0.01	1	0.05	0.05	0.04	−0.01	0.04
Coinventor	0.15	0.28	0.25	0.02	0.03	0.27	0.29	0.32	0.16	0.34	0.16	0.05	1	0.84	0.28	0.13	0.3
Three Degrees	0.17	0.28	0.23	0.03	0.06	0.27	0.29	0.3	0.16	0.37	0.16	0.05	0.84	1	0.29	0.11	0.32
Citations	0.12	0.16	0.15	0.01	0.07	0.29	0.33	0.33	0.08	0.31	0.19	0.04	0.28	0.29	1	0.06	0.47
Rare Surname	0.05	0.09	0.09	0.06	0	0.07	0.07	0.05	−0.06	0.07	0.02	−0.01	0.13	0.11	0.06	1	0.04
ASE	0.09	0.12	0.1	0	0.07	0.26	0.31	0.3	0.13	0.3	0.19	0.04	0.3	0.32	0.47	0.04	1

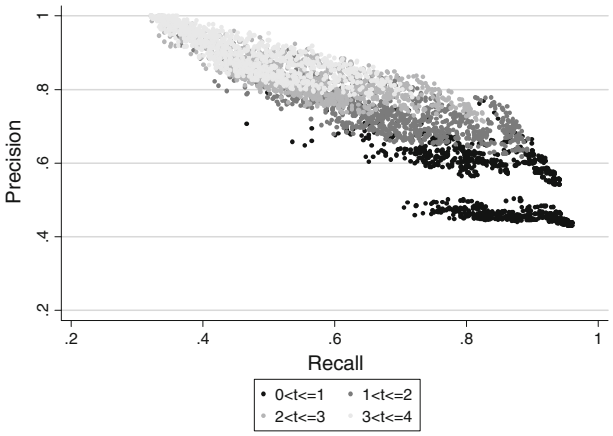

Fig. 2 Precision and Recall values according to different threshold (t) values (4,000 sets of weights)

We treat the latter as a parameter subject to calibration, too. Therefore, we add to each vector of weights ω_w, a random threshold value, extracted from a uniform distribution with upper bound 4 and lower bound zero:

$$Threshold_w = U(0, 4)$$

4. *Observations*: Each vector of weights w generates 2,817 $\alpha_{m,w}$ values in case of NAFA and 1011 $\alpha_{m,w}$ values in case of NAE, one for each inventor pair in the dataset. They come along with a threshold value ($Threshold_w$), which allows us to define $D_{\alpha_{m,w}}$ as follows.

$$D_{\alpha_{m,w}} = 1, \quad if\ \alpha_{m,w} \geq Threshold_w$$
$$D_{\alpha_{m,w}} = 0, \quad if\ \alpha_{m,w} \geq Threshold_w$$
$$m = \{m_{\mathrm{NAE}}, m_{\mathrm{NAFA}}\}$$

By comparing $D_{\alpha_{m,w}}$ and D_{γ_m} as in Eq. 3, we then compute the number of true (false) positives (negatives) obtained by applying different sets of weights and threshold values [ω_w, $Threshold_w$]. That is, we generate 4,000 records (2000 for NAE and 2000 for NAFA), to be used in our calibration exercise, each record being characterized by a different combination of precision rate, recall rate, vectors of weights and threshold value.

Figure 2 is a scatter plot for the precision and recall rates, where dots correspond to observations and dot colors indicate the relative threshold value. The figure shows the extent of the trade-off between precision and recall. It also shows how the trade-off depends on the threshold value: higher precision and lower recall for higher thresholds, and vice versa. Yet, we observe that for different threshold values we can obtain a similar combination of precision and recall, depending on the values assigned to weights w^k (overlapping regions of dots). Also Fig. 3 is a scatterplot for precision and recall rates; in this case the dots are grouped according to the benchmark databases they refer to, NAFA and NAE. We notice that NAFA dots tend to exhibit higher precision rates, given the recall

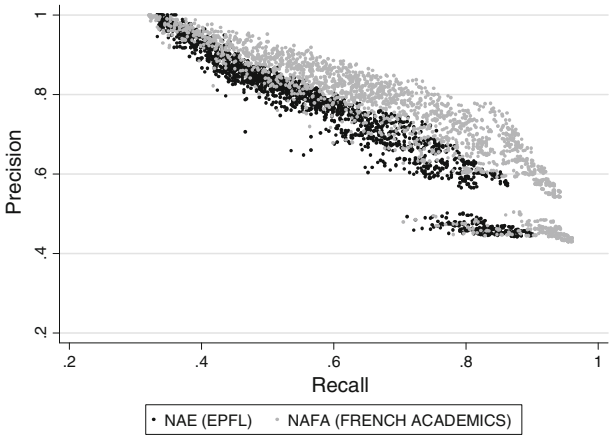

Fig. 3 Precision and Recall values according to NAFA and NAE datasets (4,000 sets of weights)

rate, and vice versa; this suggests that our algorithm fares better when applied to NAFA than to NAE, that is, it is sensitive to the benchmark chosen for calibration.

Mapping

This second step identifies the most efficient combinations of weights with respect to predefined objective regions: *high precision*, *high recall* and *balanced* mix of recall and precision levels. Outcomes (observations) in each region are first split in two groups: *dominant* and *dominated*. An outcome is dominated whenever another outcome exists which has both higher precision and higher recall; it is dominant whenever no such other outcome exists.

$$\textit{Dominant outcomes}(\bar{o}) \rightarrow \{\bar{o} : \nexists\, \underline{o}\ \text{Precision}(\underline{o}) > \text{Precision}(\bar{o}) \cap \text{Recall}(\underline{o}) > \text{Recall}(\bar{o})\}$$

$$\textit{Dominated outcomes}(\underline{o}) \rightarrow \{\underline{o} : \exists\, \bar{o}\, \text{Precision}(\underline{o}) < \text{Precision}(\bar{o}) \cap \text{Recall}(\underline{o}) < \textit{Recall}(\bar{o})\}$$

Dominant outcomes can be seen in Fig. 3 as dots at the upper frontier of the cloud of observations. If we consider separately the clouds for NAFA and NAE results, we would obtain two distinct sets of dominant outcomes, one for each benchmark dataset, as in Fig. 4.[8] Vertical lines in the figure identify nine areas, three of which include outcomes corresponding to our objectives of high precision, high recall and balanced results. In particular, the high precision area includes all dominant outcomes with precision rate higher than 0.7, and recall rate between 0.5 and 0.65; the high recall region includes all dominant outcomes with a recall rate higher than 0.65, and precision rate between 0.5 and 0.7; the balanced results region includes all dominant outcomes with a recall higher than 0.65 and precision higher than 0.7.

Notice that two other areas of potential interest are "maximum precision" and "maximum recall" (see Fig. 4). However, these are not reasonable objectives to pursue, as they

[8] The NAFA and NAE frontiers, include not only the most extreme points, but are extended to include all outcomes with precision and recall values higher than Precision($\bar{o}$)-0.02 and *Recall*($\bar{o}$)-0.02 for any $\bar{o}$. This will turn out useful for the ensuing statistical exercise.

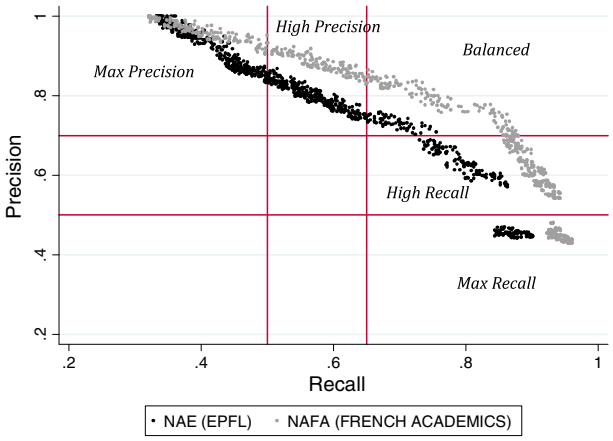

Fig. 4 Dominant Solutions for NAE and NAFA benchmarks

come at too high a cost in terms of recall and precision, respectively (e.g. to achieve max precision we should stand a recall rate of less than 0.5, which is worse than the result of just guessing).

We also calculate the number of positive weights characterizing the sets of weights within the region of interest (*AVG nr filtering criteria* positively weighted). In the *Balanced* region for the NAFA benchmark of Fig. 4, we have 132 vectors of weights, one for each algorithm run falling within the region, having on average 8.77 filtering criteria positively weighted. However, we count the number of positive weights assigned to criteria with integer numbers, then we can conclude that the 132 observations are on average characterized by nine positive weights.

Weight assignment and threshold selection

Once defined the three regions of interest, we assess which of the filtering criteria are over-represented (or under-represented) within each region, and consequently we select them for inclusion in the vector of weights representing the calibrated parametrization of the algorithm. Criterion k is over-represented (under-represented) if the expected value of its weight $E[\omega^k,]$ in the region of interest is significantly higher (lower) than 0.5.[9]

We test the over-representation (under-representation) hypothesis by means of one-tail *t tests*, with 95 % significance, as follows:

- Over-representation test for criterion k H_0: $E[\omega.^k] = 0.5$ H_1: $E[\omega.^k] > 0.5$
- Under-representation test for criterion k H_0: $E[\omega.^k] = 0.5$ H_1: $E[\omega.^k] < 0.5$

[9] Remember that ${}_w\omega^k$ is a random variable with expected value equal to 0.5. By definition, any sample with a different mean cannot be randomly drawn, and must be considered either over- or under-represented by comparison to a random distribution.In case the estimated impact of a criterion is not significantly different than zero for recall, but positive for precision, then it is desirable to include it in any parametrization, as it increases precision at no cost in terms of recall. Conversely, any filter with zero impact on precision, but significantly negative for recall, ought to be excluded from any parametrization, as it bears a cost in the terms of the latter, and no gains in terms of precision. We have conducted this type of analysis, and found it helpful to understand the relative importance of the different filtering criteria. We do not report it for reasons of space, but it is available on request.

Table 4 Averages [H_0: $E[\omega_.^k] = 0.5$ H_1: $E[\omega_.^k] > 0.5$, H_0: $E[\omega_.^k] = 0.5$ H_0: $E[\omega_.^k] < 0.5$]

NAFA (French academics)	*Balanced*			*High precision*			*High recall*		
	Mean	*P* value H_0: $E[\omega_.^k] = 0.5$ H_1: $E[\omega_.^k] > 0.5$	*P* value H_0: $E[\omega_.^k] = 0.5$ H_1: $E[\omega_.^k] < 0.5$	Mean	*P* value H_0: $E[\omega_.^k] = 0.5$ H_1: $E[\omega_.^k] > 0.5$	*P* value H_0: $E[\omega_.^k] = 0.5$ H_1: $E[\omega_.^k] < 0.5$	Mean	*P* value H_0: $E[\omega_.^k] = 0.5$ H_1: $E[\omega_.^k] > 0.5$	*P* value H_0: $E[\omega_.^k] = 0.5$ H_1: $E[\omega_.^k] < 0.5$
Network variables									
Coinventor	0.67*	*0*	*1*	0.75*	*0*	*1*	0.52*	*0.29*	*0.71*
Three Degrees	0.48*	*0.7*	*0.3*	0.67*	*0*	*1*	0.57*	*0.02*	*0.98*
Geographical variables									
City	0.42	*0.97*	*0.03*	0.29	*1*	*0*	0.47	*0.83*	*0.17*
Province	0.93*	*0*	*1*	0.71*	*0*	*1*	0.72*	*0*	*1*
Region	0.93*	*0*	*1*	0.77*	*0*	*1*	0.81*	*0*	*1*
State	0.48*	*0.64*	*0.36*	0.51*	*0.4*	*0.6*	0.46	*0.89*	*0.11*
Street	0.33	*1*	*0*	0.38	*1*	*0*	0.49	*0.61*	*0.39*
Applicant related variables									
Applicant	0.49*	*0.57*	*0.43*	0.53*	*0.22*	*0.78*	0.51*	*0.34*	*0.66*
Small Applicant	0.52*	*0.36*	*0.64*	0.41	*0.98*	*0.02*	0.56*	*0.05*	*0.95*
Group	0.52*	*0.36*	*0.64*	0.5*	*0.53*	*0.47*	0.5	*0.5*	*0.5*
Technology classes									
IPC 4	0.37	*1*	*0*	0.4	*0.99*	*0.01*	0.6*	*0*	*1*
IPC 6	0.3	*1*	*0*	0.22	*1*	*0*	0.52*	*0.25*	*0.75*
IPC 12	0.45	*0.89*	*0.11*	0.48*	*0.67*	*0.33*	0.51*	*0.39*	*0.61*
Citation related variables									
Citations	0.45	*0.89*	*0.11*	0.46	*0.83*	*0.17*	0.49	*0.66*	*0.34*
ASE	0.45	*0.89*	*0.11*	0.44	*0.91*	*0.09*	0.47	*0.79*	*0.21*
Other filtering criteria									
Rare Surname	0.6*	*0.01*	*0.99*	0.6*	*0.01*	*0.99*	0.5	*0.5*	*0.5*
Three Years	0.39	*0.99*	*0.01*	0.43	*0.93*	*0.07*	0.16	*1*	*0*

Table 4 continued

NAFA (French academics)	*Balanced*			*High precision*			*High recall*		
	Mean	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] > 0.5$	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] < 0.5$	Mean	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] > 0.5$	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] < 0.5$	Mean	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] > 0.5$	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] < 0.5$
Nr of filtering criteria and threshold									
AVG nr filtering criteria	8.77			8.57			8.86		
AVG threshold	2.22			3.16			0.76		
Observations	132			129			214		

* Positively weighted (selected) criteria

Table 5 Averages [H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] > 0.5$, H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_0: $E[\omega_{\cdot}^{k}] < 0.5$]

NAE (EPFL Scientists)	*Balanced*			*High precision*			*High recall*		
	Mean	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] > 0.5$	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] < 0.5$	Mean	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] > 0.5$	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] < 0.5$	Mean	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] > 0.5$	*P* value H_0: $E[\omega_{\cdot}^{k}] = 0.5$ H_1: $E[\omega_{\cdot}^{k}] < 0.5$
Network variable									
Coinventor	0.56*	*0.17*	*0.83*	0.51	*0.33*	*0.67*	0.54	*0.17*	*0.83*
Three Degrees	0.58*	*0.08*	*0.92*	0.54*	*0.1*	*0.9*	0.59*	*0.02*	*0.98*
Geographical variables									
City	0.6*	*0.05*	*0.95*	0.55*	*0.04*	*0.96*	0.67*	*0*	*1*
Province	0.42	*0.92*	*0.08*	0.41	*1*	*0*	0.42	*0.96*	*0.04*
Region	0.51	*0.41*	*0.59*	0.47	*0.84*	*0.16*	0.42	*0.96*	*0.04*
State	0.46	*0.76*	*0.24*	0.54	*0.08*	*0.92*	0.56*	*0.1*	*0.9*
Street	0.42	*0.92*	*0.08*	0.38	*1*	*0*	0.56*	*0.1*	*0.9*
Applicant related variables									
Applicant	0.94*	*0*	*1*	0.75*	*0*	*1*	0.93*	*0*	*1*
Small Applicant	0.63*	*0.02*	*0.98*	0.49	*0.67*	*0.33*	0.64*	*0*	*1*
Group	0.43	*0.88*	*0.12*	0.46	*0.92*	*0.08*	0.53	*0.27*	*0.73*
Technology classes									
IPC 4	0.38	*0.98*	*0.02*	0.62*	*0*	*1*	0.75*	*0*	*1*
IPC 6	0.38	*0.98*	*0.02*	0.52*	*0.26*	*0.74*	0.56*	*0.1*	*0.9*
IPC 12	0.69*	*0*	*1*	0.65*	*0*	*1*	0.56	*0.07*	*0.93*
Citation related variables									
Citations	0.54*	*0.24*	*0.76*	0.59*	*0*	*1*	0.52	*0.33*	*0.67*
ASE	0.53*	*0.32*	*0.68*	0.52*	*0.26*	*0.74*	0.56*	*0.1*	*0.9*
Other filtering criteria									
Rare Surname	0.28	*1*	*0*	0.5	*0.46*	*0.54*	0.41	*0.98*	*0.02*
Three Years	0.83*	*0*	*1*	0.65*	*0*	*1*	0.23	*1*	*0*

Table 5 continued

NAE (EPFL Scientists)	*Balanced*			*High precision*			*High recall*		
	Mean	*P* value H_0: $E[\omega_\cdot^k] = 0.5$ H_1: $E[\omega_\cdot^k] > 0.5$	*P* value H_0: $E[\omega_\cdot^k] = 0.5$ H_1: $E[\omega_\cdot^k] < 0.5$	Mean	*P* value H_0: $E[\omega_\cdot^k] = 0.5$ H_1: $E[\omega_\cdot^k] > 0.5$	*P* value H_0: $E[\omega_\cdot^k] = 0.5$ H_1: $E[\omega_\cdot^k] < 0.5$	Mean	*P* value H_0: $E[\omega_\cdot^k] = 0.5$ H_1: $E[\omega_\cdot^k] > 0.5$	*P* value H_0: $E[\omega_\cdot^k] = 0.5$ H_1: $E[\omega_\cdot^k] < 0.5$
Nr of filtering criteria and threshold									
AVG nr filtering criteria	9.17			9.15			9.43		
AVG threshold	1.42			2.42			0.75		
Observations	72			336			135		

* Positively weighted (selected) criteria

We then proceed by including in the algorithm all over-represented criteria (that is, we assign them weight $\omega^k = 1$), and excluding the under-represented ones (assign $\omega^k = 0$), depending on the objective region.

Tables 4 and 5 report for each filtering criterion, the sample mean of its weight and the p values of the one-tail *t tests*. Separate tests are run for NAFA and NAE benchmark datasets and for the three regions of interest.

For illustration consider *City* and *State* criteria from Table 4 (NAFA dataset) in the *Balanced* precision-recall region.[10]

We observe a sample mean equal to 0.42 for *City* criterion, which translates into a rejection of the null hypothesis in the under-representation test (p value = 0.03), but not in the over-representation test (p value = 0.97). As the *City* criterion is significantly under-represented in the observations characterized by *Balanced* objective, then we exclude it by assigning to *city* criterion a zero-weight ($\omega^{City} = 0$).

On the contrary, for the *State* criterion, the null hypothesis cannot be rejected either in the under-representation test nor in the over-representation test (p-values being respectively 0.64 and 0.36). This means that *t tests* do not give a clear (and statistically significant) evidence to help us deciding whether to include or exclude the *State* criterion. In this case we give a positive weight to the *State* criterion only if it contributes to reach, in the calibrated parametrization, the average number of positively weighted filtering criteria characterizing the observations in the objective region. (that is, if the positively weighted criteria selected in the calibrated parametrization are less than the nine observed on average in the *Balanced* case, *State* is included).

Results of the tests provide us with guidance for choosing the filtering criteria to include (assign positive weight) in calibrated parametrization of the algorithm, according to the precision and recall objectives we aim at. For sake of simplicity we identify the positively weighted criteria with an asterisk in Tables 4 and 5.

In the case of NAFA benchmark, whatever the objective region, *network* criteria are always assigned a positive weight. In case of NAE only *three degrees* is assigned a positive weight, for all the three regions.

The family of *geographic* criteria plays an important role in the NAFA benchmark, but not in the NAE benchmark. This is not surprising given the low quality of geographical information for Swiss inventors available on PatStat data (see Lissoni et al. 2010). *Applicant* and *Technology* families show a mixed evidence, the choice of weights being specific to any combination of benchmark dataset and objective regions. The *Citation* family does not play any role in NAFA dataset, while it has to be weighted positively in NAE dataset. Among the remaining criteria (*others* family), having a *rare surname* has to be included in NAFA database when objective regions are *Balanced* and *High precision*, as well as *three years* in case of NAE benchmark database.

Once defined the vector of weights for the calibrated parametrization of the algorithm, a threshold value is needed, which we calculate as the average threshold value within each region. For instance, in the *balanced* region of the NAFA benchmark, the average threshold value for the 132 outcomes (dots) is 2.22. It means that the similarity score $\alpha_{m,w}$ must be equal to or higher than 2.22 for any match to be considered positive. As expected,

[10] Regression analysis can be applied to the same set of results in order to estimate the marginal impact of each filtering criterion and the *Threshold* on either precision and recall, other things being equal. In general, we expect all filters to bear a negative influence on recall (in that they increase the number of negative matches, both true and false), and a positive influence on precision (they eliminate false positives).

the average threshold value is highest in the high precision region and lowest in the high recall one (see Tables 4 and 5).

Validation and application to PatStat data

Following our calibration exercise, we produce three versions (parametrizations) of Massacrator 2.0, one for each precision-recall objective, with weights and threshold calculated accordingly. We then check to what extent each of these parametrizations is satisfying in terms of the precision and recall rates it produces, conditional on its objective. Precision and recall rates are measured, once again, against the NAFA and NAE benchmarks.

Parametrization

We run each version of Massacrator©, once for each combination for each benchmark, for a total of 6 six runs, with the following results:

- NAFA dataset-precision oriented parametrization→Precision:92 % Recall:54 %.
- NAFA dataset-recall oriented parametrization→Precision:56 % Recall:93 %.
- NAFA dataset-balanced parametrization→Precision:88 % Recall:68 %.
- NAE dataset-precision oriented parametrization→Precision:79 % Recall:62 %.
- NAE dataset-recall oriented parametrization→Precision:59 % Recall:85 %.
- NAE dataset-balanced parametrization→Precision:74 % Recall:70 %.

Notice that by calibrating our filtering step on either NAE or NAFA we obtain different results. This is because each dataset has a number of semantic peculiarities (variety of names and; quality of information contained in the addresses; variety in the technological classes and citations of patents), which are mirrored by differences in the number and type of criteria selected at the calibration stage.

This forced us to choose only one benchmark dataset to perform our final calibration, the one leading to the production of the APE-INV dataset. Our choice fell on NAFA, which contains higher quality information for addresses, and more name variety. For the three alternative parametrizations of Massacrator algorithm we then obtain the following disambiguation results[11]:

- NAFA calibrated, precision-oriented parametrization: from 3,896,945 inventors in the original PatStat database (for EPO patents) we obtain 3,662,515 disambiguated inventors (unique codes) in APE-INV, that is −7 %.
- NAFA calibrated, recall-oriented parametrization: from 3,896,945 inventors to 2,210,277 unique codes, that is −44 %.
- NAFA calibrated, balanced parametrization: from 3,896,945 inventors to 3,474,891 unique codes, that is −11 %.

[11] The figures presented here are the result of further adjustments we introduced in order to solve transitivity problems. Transitivity problems may emerge for any triplet of inventors (such as I, J, and Z) whenever two distinct pairs are recognized to be same person (e,g, I & J and J and Z), but the same does not apply to the remaining pair (I & Z are not matched, or are considered negative matches). In this case we need to decide whether to revise the status of I & Z (and consider the two inventors as the same person as J) or the status of the other pairs (and consider either I or Z as different persons than J). When confronting this problem, we always opted for considering the two inventors the same person, then I,J and Z are the same individual according to Massacrator.

Table 6 Descriptive statistics of inventorship: Massacrator runs with different parametrizations on the whole PatStat dataset

	(1)	(2)	(3)	(4)	(5)	(6)
Parametrization	Avg Patent per inventor	Star inventors' productivity	International mobility index	Connected nodes %	Centralization-degree %	Density %
Balanced	2.1705	3.56 %	1.02 %	95 %	0.156	0.0034
Recall-oriented	3.0244	8.19 %	4.92 %	95.21 %	0.515	0.0053
Precision-oriented	2.0381	3.48 %	0.56 %	95 %	0.149	0.0032

As expected the largest reduction in the number of inventors is obtained with the recall-oriented parametrization, the smallest with the precision-oriented one. More importantly, when applying data disambiguated with different precision-recall objectives to classic problems in the economics of innovation or science and technology studies, we will get different results. As an illustration, consider three classical topics: inventors' productivity, mobility, and social networking (on the latter topic, see Borgatti et al. 2009 for technical vocabulary and basic concepts). Table 6 reports descriptive statistics for each topic, as resulting from datasets built by using different parametrizations of Massacrator, namely:

- *Avg. Patent per inventor*: it is the average number of patents per inventor in the whole dataset.
- *Star inventors' productivity*: it is the share of patents belonging to the 1,000 most prolific inventors in the database.
- *International mobility index*: It is the share of inventors with at least two different country addresses, over the total number of inventors with at least two patents (inventors with only one patent are not considered, as they can have only one address, by definition).
- *Connectedness*: it is the percentage of connected nodes over the total number of nodes in the network of inventors active between 2000 and 2005 in the fields of chemistry and pharmaceuticals (from now on: Net2000/05)[12] Isolated nodes represent individuals with no co-inventorship relationships over the period considered.
- *Centralization-degree*: it is a degree-based measure of graph centrality for Net2000/05, as defined in Freeman (1979). It measures the extent at which the graph structure is organized around focal points, and it reaches a maximum value for a star graph.
- *Density*: it is the number of observed ties in Net2000/05, over the maximum number of possible ties (i.e. the number of ties in a fully connected network with the same number of nodes). It measures the intensity of connections between inventors.

As expected the productivity index in column (1) is higher for the recall-oriented parametrization of the algorithm, on the basis of which we treat a larger number of inventors as the same individual. As similar consideration is valid also for statistics on star inventors, which are assigned a maximum of 8 % of patents when using a recall-oriented parametrization and only 3.5 % with a precision-oriented parametrization. As for international mobility, its index ranges from 0.56 % to 4.92 % according to the parametrization

[12] Fields of chemistry and pharmaceuticals are defined as in Schmoch (2008). We consider only these fields, and years from 2000 and 2005, for ease of computation. Co-inventorship is intended as a connection between two inventors having (at least) one patent in common.

choice. While productivity measures do not change much when moving from the *Precision*-oriented parametrization to the *Balanced* one, the same cannot be said for mobility measures: in this case, even the modest reduction in precision (increase in recall) introduced by changing algorithm changes considerably the value of the indicator.

As for network measures, *Connectedness* is not very sensitive to the algorithm parametrization. The same cannot be said for *Centralization* and *Density*, both of which increase considerably along with recall and decline with precision.

Language- and country-specific issues

An important limitation of Massacrator 2.0 (one shared with most available algorithms) lies in the generality of the matching rules for names and surnames, as well as in its dependence, for the calibration of filtering criteria, on a limited set of training (benchmark) data.

Concerning matching criteria, 2G distances tend to be lower when names and surnames tend to be shorter, as it is the case in several Asian countries (for example, South Korea and China; but not Japan). In addition, the probability of matching two entities is higher when a small number of names and/or surname tend to exhibit a very high frequency within the population (as it is the case with Asian countries, this time including Japan, too; and, in Europe, with Scandinavian countries).

As for filtering, the two datasets used for the calibration of Massacrator 2.0, namely NAFA and NAE, are country specific (respectively, for France and Switzerland). At first sight, this may suggest that the filtering weights resulting from the parametrization exercise may also be affected by several language- or country-specific items in the benchmarks, such as, again, the relative frequency of names and surnames (and the average lexical distance between them), or the concentration of the population in large cities. Notice however that, of the 17 filtering criteria considered, only five depend on geography (co-localization of the matched entities) and just one on the frequency of surnames. All the other criteria make use of information contained in the patents, such as citations, technological classification, and co-inventorship.

Ideally, we would like to assemble as many training datasets as the number of inventors' countries we wish to investigate, and to produce as many versions (parametrizations) of the Massacrator 2.0 algorithm. At this stage, however, a similar effort would be prohibitively costly. Besides, it would not solve entirely the problem, as inventors in a certain number of high-immigration countries, such as the US, Canada, Australia, or several small European countries (such as Switzerland or the Netherlands) have names and surnames that respond to different linguistic rules (and we do not know in advance to which linguistic group they belong to).[13]

In the absence of a ready solution to these problems, we want at least to quantify the bias induced by the limitations of our training datasets, for the inventors whose countries of residence (at the time of the patent filing) coincided with one of the ten countries with the largest number of patent applications filed at the EPO (namely: US, Japan, Germany, France, Great Britain, Italy, South Korea, The Netherlands, Switzerland, and Canada). Table 7 reports the total number of original inventor occurrences for such countries

[13] On immigration of inventors, see Miguelez and Fink (2013) and Breschi et al. (2014). Both papers provide information on ongoing attempts to classify inventors according to their nationality and/or country of origin (country of birth, or of parents' or grandparents' birth). In the near future, it will be possible to use such information to refine new versions of Massacrator (see Conclusions).

Table 7 Country specific consequences of applying the Massacrator algorithm (balanced calibration)

Country of residence*	Nr of inventors	% of pairs (matching step)	% of confirmed pairs (filtering step)	% reduction of nr of inventors due to disambiguation
United States	908,163	0.00051 %	13 %	−21 %
Japan	750,534	0.00206 %	2 %	−12 %
Germany	445,614	0.00075 %	13 %	−17 %
France	198,314	0.00089 %	24 %	−15 %
Great Britain	171,203	0.00249 %	25 %	−29 %
Italy	83,836	0.00148 %	52 %	−21 %
South Korea	82,794	0.00891 %	5 %	−12 %
The Netherlands	69,843	0.00229 %	7 %	−5 %
Switzerland	57,738	0.00225 %	16 %	−9 %
Canada	50,968	0.00334 %	19 %	−13 %

* At the time of the patent filing

(Column 2), as well as the reduction in terms of number of disambiguated inventors when we apply the balanced algorithm (Column 5). Columns 3 and 4 show two intermediate results of the algorithm at matching and filtering step. We measure the matching result as the ratio of the number of pairs of inventors generated during the matching step divided by the total number of possible combination of inventors within the country (Eq. 4 defines the values in column 3). Then, we measure the outcome of the filtering step as share of pairs confirmed by the filtering criteria (Eq. 5 defines the values in column 4).

$$\%\, of\ pairs = \frac{nr\ of\ pairs}{((nr\ of\ inventors) * (nr\ of\ inventors\ -\ 1)/2)} \tag{4}$$

$$\%\, of\ confirmed\ pairs = \frac{nr\ of\ confirmed\ pairs}{nr\ of\ pairs} \tag{5}$$

Not surprisingly we find that we are more likely to observe a proliferation of pairs in Korea and partially in Japan, whose variety of names and surnames is rather low, for historical reasons (namely, the low level of immigration, the absence of multiple ethnic minorities, and several linguistic peculiarities; see Yasuda, 1983). For these countries the filtering criteria play a major role, as it helps rejecting many false positive (it confirms only a small share of the inventors matched, respectively 2 and 5 %). In some western countries, on the contrary, the number of pairs generated in the matching step is lower (higher heterogeneity of names and surnames), and the share of pairs confirmed by filters is notably higher. For instance, Italy (whose variety of surnames is notoriously high; Barrai et al. 1999) shows a low percentage of pairs (0.00148 %) and the highest share of confirmed pairs (52 %) in the filtering step: one out of two pairs identified by the matching step is confirmed by the filtering step. The same applies to France, whose population is large, but has a long story of immigration, as well as high regional variety in surnames.

All in all, evidence in Table 7 suggests that our matching rules resent heavily of language- and country-specific biases, but also that our filtering rules do quite a good job in countering this bias, by validating only a small percentage of matches for countries where we observe many of them. As a result, the order of magnitude in the reduction of the number of entities (inventors) due to the disambiguation exercise does not vary much

across countries. The only puzzling case is given by the Netherlands where the number of pairs generated during the matching step is quite high and the percentage of confirmed pairs is very low (5 %). It results in the smallest reduction of the number of entities during the disambiguation exercise.

Conclusions and further research

In this paper we have presented a general methodology for inventor disambiguation, with an application to EPO patent data. We have argued that producing high quality data requires calibrating the choice of weights by means of simulation analysis. Calibration is necessary to:

1. identify "frontier" results, that is the set of efficient weights that maximise the precision rate, conditional on a given recall rate (or, vice versa, recall conditional on precisions); in this way, one excludes inefficient sets of weights and make less arbitrary choices;
2. allow the researcher to choose between precision-oriented and recall-oriented algorithms, or to combine them into a a balanced one.

Choosing one algorithm over the others may be desirable when the research purposes require minimization of either errors of type I or errors of type II (respectively, false positives and false negatives). For example, early research on academic patenting by Lissoni et al. (2008) was aimed at proving that official estimates of the number of academic patents (namely, patents signed by at academic scientists) in Europe were wrong by defect, and thus needed to minimize errors of type I. A more recent study on the same topic, on the contrary, has produced a longitudinal database of academic patenting in Italy, with the primary objective of detecting trends Lissoni et al. (2013). With that objective in mind, there is no reason to prefer minimization of either errors of type I or errors of type II, so the authors make use of the APE-INV inventor database described in this paper, with balanced parametrization.

More generally, we have shown how different calibrations lead to different results for the fundamental indicators of studies on inventors' productivity, mobility, and networking. This means that the results of several studies recently published on these topics, which do not provide details on the disambiguation methods they followed, could turn out not be robust to the calibration choices presented here (this include some work by one us, such as: Balconi et al. 2004 on networks; or Breschi and Lissoni 2009, on mobility). For sure, future research results in these area will have to be screened more closely, and disambiguation methods made explicit.

Besides conducting robustness test for different disambiguation parametrizations, authors may also pursue the road of combining the results of different calibrations in order to increase data quality. This can be done by comparing the results, for each pair of records, of the different calibrations and choose on the basis of whether a majority of the algorithms suggest the same results. The combination principles can be extended not only to different calibration algorithms, but to altogether different algorithms, as discussed by Maurino et al. (2012).

One last strategy for further data quality improvements can consist in sharing more openly inventor data and collecting feedbacks from other users. This is an integral part of the APE-INV project, for which the inventor database described in this paper was produced and made available online (http://www.ape-inv.disco.unimib.it/). Users who check

manually the inventor data they download, or match them to other sources of information on individuals (such as lists of academics or authors of scientific papers) do inevitably find a number of false negatives or false positives. The same holds if their research requires contacting the inventors for interview or survey purposes. This user-generated information is extremely valuable, and we believe it is worth investing in finding ways to collect it. To this end we have set up the APE-INV User's Feedback project, which invites users to come back to the APE-INV data website and upload either their proposed corrections to the APE-INV inventor dataset, or the results of their own disambiguation exercises based on the same set of data (for a full description of the project, see Den Besten et al. 2012).

Users may also play a role in assembling country- and language-specific training (benchmark) datasets, which would possibly result in as many specific versions of the algorithms, so to overcome the unequal cross-country distribution of precision and recall rates, of which we suspect the results of Massacrator 2.0 to be affected. Waiting for such data to be made available, it is worth exploring the possibility to improve the algorithm by bringing in information on nationality or country of origin of inventors, from the sources mentioned in section Language- and country-specific issues.

Acknowledgements This paper derives from research undertaken with the support of APE-INV, the Research Networking Programme on Academic Patenting in Europe, funded by the European Science Foundation.. Early drafts of benefitted from comments by participants to the APE-INV NameGame workshop series. We are also grateful to Nicolas Carayol, Lorenzo Cassi, Stephan Lhuillery and Julio Raffo for providing us with core data for the two benchmark datasets. Monica Coffano and Ernest Miguelez provided extremely valuable research assistantship. Andrea Maurino's expertise on data quality has been extremely helpful.

References

Agrawal, A., Cockburn, I., & McHale, J. (2006). Gone but not forgotten: knowledge flows, labor mobility, and enduring social relationships. *Journal of Economic Geography, 6*(5), 571.

Azoulay, P., Ding, W., & Stuart, T. (2009). The impact of academic patenting on the rate, quality and direction of (public) research output. *The Journal of Industrial Economics, 57*, 637–676.

Balconi, M., Breschi, S., & Lissoni, F. (2004). Networks of inventors and the role of academia: an exploration of Italian patent data. *Research Policy, 33*(1), 127–145.

Barrai, I., Rodriguez-Larralde, A., Mamolini, E., & Scapoli, C. (1999). Isonymy and isolation by distance in Italy. *Human biology, 71*, 947–961.

Bilenko, M., Kamath, B., & Mooney, R.J. (2006). Adaptive blocking: Learning to scale up record linkage, In Data Mining, 2006. ICDM'06. Sixth International Conference on. IEEE, pp. 87–96.

Borgatti, S. P., Mehra, A., Brass, D. J., & Labianca, G. (2009). Network analysis in the social sciences. *science, 323*(5916), 892–895.

Breschi, S., & Lissoni, F. (2005). Knowledge networks from patent data. In H. F. Moed, W. Glänzel & U. Schmoch (Eds.), *Handbook of Quantitative Science and Research*. Amsterdam: Springer.

Breschi S., & Lissoni F. (2009). Mobility of skilled workers and co-invention networks: an anatomy of localized knowledge flows. Journal of Economic Geography.

Breschi, S., Lissoni, F., & Montobbio, F. (2008). University patenting and scientific productivity: a quantitative study of Italian academic inventors. *European Management Review, 5*(2), 91–109.

Breschi, S., Lissoni, F., & Tarasconi, G. (2014). Inventor Data for Research on Migration & Innovation: a Survey and a Pilot. *WIPO Economic Research Working Paper*. N.17, World Intellectual Property Organization, Geneva.

Burt, R. S. (1987). Social contagion and innovation: cohesion versus structural equivalence. American journal of Sociology, 1287–1335.

Carayol, N., & Cassi, L. (2009). Who's Who in Patents. A Bayesian approach. *Cahiers du GREThA, 7*, 07–2009.

Den Besten M., Lissoni F., Maurino A., Pezzoni M., & Tarasconi G. (2012). Ape-Inv Data Dissemination And Users' Feedback Project", mimeo (http://www.academicpatentig.eu).

Fleming, L., King, C., & Juda, A. I. (2007). Small Worlds and Regional Innovation. *Organization Science, 18*, 938–954.

Freeman, L. C. (1979). Centrality in social networks conceptual clarification. *Social Networks, 1*(3), 215–239.

Griliches, Z. (1990). Patent statistics as economic indicators: A survey. *Journal of Economic Literature,28*(4), 1661–1707.

Huang, H., & Walsh, J. P. (2011). A new name-matching approach for searching patent inventors. mimeo.

Li, G.C., Lai, R., D'Amour, A., Doolin, D.M., Sun, Y., Torvik, V.I., Yu, A.Z., & Fleming, L. (2014). Disambiguation and co-authorship networks of the US patent inventor database. *Research Policy, 43*(6), 941–955.

Lissoni, F., Coffano, M., Maurino, A., Pezzoni, M., & Tarasconi, G. (2010). APE-INV's Name Game Algorithm Challenge: A Guideline for Benchmark Data Analysis & Reporting. mimeo.

Lissoni, F., Llerena, P., McKelvey, M., & Sanditov, B. (2008). Academic patenting in Europe: new evidence from the KEINS database. *Research Evaluation, 17*(2), 87–102.

Lissoni, F., Pezzoni, M., Poti, B., & Romagnosi, S. (2013). University Autonomy, the Professor Privilege and Academic Patenting: Italy, 1996–1997. *Industry and Innovation, 20*(5), 399–421.

Lissoni, F., Sanditov, B., & Tarasconi, G. (2006). The Keins database on academic inventors: methodology and contents. WP cespri, 181.

Marx, M., Strumsky, D., & Fleming, L. (2009). Mobility, skills, and the Michigan non-compete experiment. *Management Science, 55*(6), 875–889.

Maurino A., Li P. (2012). Deduplication of large personal database. Mimeo.

Miguelez, E., & Fink, C. (2013). Measuring the International Mobility of Inventors: A New Database, *WIPO Economic Research Working Paper* N.8, World Intellectual Property Organization, Geneva.

Nagaoka, S., Motohashi, K., & Goto, A. (2010). Patent statistics as an innovation indicator. *Handbook of the Economics of Innovation, 2*, 1083–1127.

On, B.-W., Lee, D., Kang, J., & Mitra, P. (2005). Comparative study of name disambiguation problem using a scalable blocking-based framework, In: Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries. ACM, pp. 344–353.

Raffo, J., & Lhuillery, S. (2009). How to play the name game: patent retrieval comparing different heuristics. *Research Policy, 38*(10), 1617–1627.

Schmoch, U. (2008). Concept of a technology classification for country comparisons. Final report to the World Intellectual Property Organization (WIPO), Fraunhofer Institute for Systems and Innovation Research, Karlsruhe.

Smalheiser, N. R., & Torvik, V. I. (2009). Author name disambiguation. *Annual review of information science and technology, 4*(1), 31–43.

Torvik, V. I., & Smalheiser, N. R. (2009). Author name disambiguation in MEDLINE. *ACM Transactions on Knowledge Discovery from Data, 3*(3), 1–29. doi:10.1145/1552303.1552304.

Torvik, V. I., Weeber, M., Swanson, D. R., & Smalheiser, N. R. (2005). A probabilistic similarity metric for Medline records: A model for author name disambiguation. *Journal of the American Society for Information Science and Technology, 56*(2), 140–158. doi:10.1002/asi.20105.

Yasuda, N. (1983). Studies of isonymy and inbreeding in Japan. *Human biology*, 263–276.